

Docket No.: 57454-132

PATENT

#3
9-14-01
SM
JE903 U.S. PTO
09/874239
06/05/01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Akira YAMADA

Serial No.:

Group Art Unit:

Filed: June 6, 2001

Examiner:

For: SYNCHRONOUS SIGNAL PRODUCING CIRCUIT FOR CONTROLLING A DATA
READY SIGNAL INDICATIVE OF END ACCESS TO A SHARED MEMORY AND
THEREBY CONTROLLING SYNCHRONIZATION BETWEEN PROCESSOR AND
COPROCESSOR

**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Commissioner for Patents
Washington, DC 20231

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicant hereby claims the priority of:

Japanese Patent Application No. 2000-223593,
Filed July 25, 2000

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Stephen A. Becker

Registration No. 26,527

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 SAB:ykg
Date: June 6, 2001
Facsimile: (202) 756-8087

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

57454-132

June 6, 2000

YAMADA

McDermott, Will & Emery

U.S. PRO
09/874239



別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

2000年 7月25日

願番号
Application Number:

特願2000-223593

願人
Applicant(s):

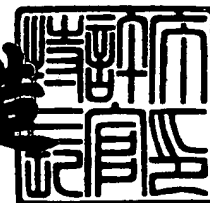
三菱電機株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 9月 1日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3070241

【書類名】 特許願

【整理番号】 525115JP01

【提出日】 平成12年 7月25日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/02
G06F 9/34
G06F 9/38
G06F 9/46

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 山田 朗

【特許出願人】

【識別番号】 000006013

【氏名又は名称】 三菱電機株式会社

【代理人】

【識別番号】 100064746

【弁理士】

【氏名又は名称】 深見 久郎

【選任した代理人】

【識別番号】 100085132

【弁理士】

【氏名又は名称】 森田 俊雄

【選任した代理人】

【識別番号】 100091409

【弁理士】

【氏名又は名称】 伊藤 英彦

【選任した代理人】

【識別番号】 100096781

【弁理士】

【氏名又は名称】 堀井 豊

【選任した代理人】

【識別番号】 100096792

【弁理士】

【氏名又は名称】 森下 八郎

【手数料の表示】

【予納台帳番号】 008693

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 同期信号生成回路、それを用いたプロセッサシステムおよび同期信号生成方法

【特許請求の範囲】

【請求項 1】 プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路であって、

前記共有メモリにおける前記プロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、

前記プロセッサが前記アクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、

前記コプロセッサがコプロセッサ命令を実行中であることを示す信号および前記比較回路の比較結果に基づいて、前記プロセッサをウェイト状態にする信号を出力する第 1 の論理回路とを含む、同期信号生成回路。

【請求項 2】 前記第 1 の論理回路は、前記コプロセッサがコプロセッサ命令を実行中であることを示す信号、前記比較回路の比較結果および共有メモリをロック状態にしないことを示す信号に基づいて、前記プロセッサをウェイト状態にする信号を出力する、請求項 1 記載の同期信号生成回路。

【請求項 3】 前記同期信号生成回路はさらに、バスウェイトサイクル数をカウントするバスウェイトカウンタと、

前記コプロセッサがコプロセッサ命令を実行中であることを示す信号、前記比較回路の比較結果およびバスウェイトカウンタのカウント値に基づいて、前記プロセッサにバスエラー信号を出力する第 2 の論理回路とを含む、請求項 1 または 2 記載の同期信号生成回路。

【請求項 4】 前記同期信号生成回路はさらに、前記第 1 の論理回路が出力する信号によって前記プロセッサがウェイト状態となるときウェイト数を設定する第 1 の初期値レジスタと、

前記プロセッサの通常アクセス時におけるウェイト数を設定する第 2 の初期値レジスタと、

前記コプロセッサがコプロセッサ命令を実行中であることを示す信号および前

記比較回路の比較結果に基づいて、前記第 1 の初期値レジスタの値および前記第 2 の初期値レジスタの値を選択して出力するセレクトと、

前記セレクトによって選択された値を入力し、カウントしたウェイトサイクル数が前記選択された値となったときに前記プロセッサにバスエラー信号を出力するバスウェイトカウンタとを含む、請求項 1 または 2 記載の同期信号生成回路。

【請求項 5】 前記同期信号生成回路はさらに、前記コプロセッサがコプロセッサ命令を実行中であることを示す信号および前記比較回路の比較結果に基づいて、前記プロセッサを低消費電力モードにする信号を出力する第 3 の論理回路を含む、請求項 1 ～ 4 のいずれかに記載の同期信号生成回路。

【請求項 6】 プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路であって、

前記共有メモリにおける前記プロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、

前記プロセッサが前記アクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、

前記コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持するコプロセッサ命令実行フラグと、

前記コプロセッサ命令実行フラグに保持される情報および前記比較回路の比較結果に基づいて、前記プロセッサをウェイト状態にする信号を出力する論理回路とを含む、同期信号生成回路。

【請求項 7】 前記アクセス禁止領域レジスタは、前記共有メモリにおける前記プロセッサの第 1 のアクセス禁止領域を指定する第 1 のアクセス禁止領域レジスタと、前記共有メモリにおける前記プロセッサの第 2 のアクセス禁止領域を指定する第 2 のアクセス禁止領域レジスタとを含み、

前記比較回路は、前記プロセッサが前記第 1 のアクセス禁止領域レジスタに指定された第 1 のアクセス禁止領域にアクセスしたことを検出する第 1 の比較回路と、前記プロセッサが前記第 2 のアクセス禁止領域レジスタに指定された第 2 のアクセス禁止領域にアクセスしたことを検出する第 2 の比較回路とを含み、

前記コプロセッサ命令実行フラグは、前記コプロセッサがコプロセッサ命令を

実行中であることを示す情報を保持する第1のコプロセッサ命令実行フラグと、
第2のコプロセッサ命令実行フラグとを含み、

前記論理回路は、前記第1のコプロセッサ命令実行フラグに保持される情報および前記第1の比較回路の比較結果に基づいて、前記プロセッサをウェイト状態にする信号を出力する第1の論理回路と、前記第2のコプロセッサ命令実行フラグに保持される情報および前記第2の比較回路の比較結果に基づいて、前記プロセッサをウェイト状態にする信号を出力する第2の論理回路とを含む、請求項6記載の同期信号生成回路。

【請求項8】 プロセッサと、

コプロセッサと、

前記プロセッサおよび前記コプロセッサに接続される共有メモリと、

前記プロセッサおよび前記コプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路とを含むプロセッサシステムであって、

前記同期信号生成回路は、前記共有メモリにおける前記プロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、

前記プロセッサが前記アクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、

前記コプロセッサがコプロセッサ命令を実行中であることを示す信号および前記比較回路の比較結果に基づいて、前記プロセッサをウェイト状態にする信号を出力する論理回路とを含む、プロセッサシステム。

【請求項9】 プロセッサと、

コプロセッサと、

前記プロセッサおよび前記コプロセッサに接続される共有メモリと、

前記プロセッサおよび前記コプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路とを含むプロセッサシステムであって、

前記同期信号生成回路は、前記共有メモリにおける前記プロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、

前記プロセッサが前記アクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、

前記コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持するコプロセッサ命令実行フラグと、

前記コプロセッサ命令実行フラグに保持される情報および前記比較回路の比較結果に基づいて、前記プロセッサをウェイト状態にする信号を出力する論理回路とを含む、プロセッサシステム。

【請求項 1 0】 プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成方法であって、

前記共有メモリにおける前記プロセッサのアクセス禁止領域を指定するステップと、

前記プロセッサが前記指定されたアクセス禁止領域にアクセスしたことを検出するステップと、

前記コプロセッサがコプロセッサ命令を実行中であることを示す信号および前記検出結果に基づいて、前記プロセッサをウェイト状態にする信号を生成するステップとを含む、同期信号生成方法。

【請求項 1 1】 プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成方法であって、

前記共有メモリにおける前記プロセッサのアクセス禁止領域を指定するステップと、

前記プロセッサが前記指定されたアクセス禁止領域にアクセスしたことを検出するステップと、

前記コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持するステップと、

前記保持される情報および前記検出結果に基づいて、前記プロセッサをウェイト状態にする信号を生成するステップとを含む、同期信号生成方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、プロセッサとコプロセッサとの間の同期を確立する技術に関し、特に、共有メモリへのアクセスの終了を示すデータレディ信号を制御してプロセッ

サとコプロセッサとの間の同期を制御する同期信号生成回路、それを用いたプロセッサシステムおよび同期方法に関する。

【0002】

【従来の技術】

近年、パーソナルコンピュータ等の情報処理装置や家電機器などに高速プロセッサが搭載されているが、処理速度の向上に対する要望は益々大きくなりつつある。これらの処理速度の向上を図る手法の1つとして、プロセッサ以外にコプロセッサを搭載し、プロセッサが苦手とする複雑な演算処理をコプロセッサに行わせる手法を挙げることができる。プロセッサとコプロセッサとの間のデータの授受は、プロセッサとコプロセッサとの双方からアクセス可能な共有メモリを用いて行うのが、データ転送処理を少なくする点において有効である。

【0003】

プロセッサとコプロセッサとを用いたシステムにおいては、プロセッサがコプロセッサに対してコプロセッサ命令を発行する。そして、コプロセッサがコプロセッサ命令を実行後、実行結果を共有メモリに格納する。コプロセッサがコプロセッサ命令を実行中は、プロセッサはそれと並行して他の処理を行なう。プロセッサがコプロセッサ命令の実行結果を利用するためには、コプロセッサ命令の完了後に、プロセッサがコプロセッサ命令の実行結果が格納される共有メモリの領域にアクセスする必要がある。このことを、本明細書において「同期」または「同期を取る」と表現することにする。

【0004】

コプロセッサ命令の完了前に、プロセッサがコプロセッサ命令の実行結果が格納されるべき共有メモリの領域のデータを使用すると、誤った処理を行うことになる。この誤った処理を避けるために、プロセッサがコプロセッサ命令の処理状況を示すコプロセッサ内の状態フラグをポーリングしたり、コプロセッサからの割込み要求を受付けたりして、コプロセッサ命令の完了を知ることによって、プロセッサとコプロセッサとの間の同期が取られている。

【0005】

【発明が解決しようとする課題】

しかし、上述したコプロセッサ内の状態フラグをポーリングする方法においては、ポーリングのためのプロセッサによる無駄な処理サイクルが必要となり、プロセッサの処理速度が低下するという問題点があった。また、コプロセッサからの割込み要求を受付ける方法においても、割込み処理のためのオーバーヘッド、たとえば、レジスタの退避／復帰などが必要となり、プロセッサの処理速度が低下するという問題点があった。

【0006】

本発明は、上記問題点を解決するためになされたものであり、第1の目的は、プロセッサとコプロセッサとの同期を取るときのオーバーヘッドを低減した同期信号生成回路およびその方法を提供することである。

【0007】

第2の目的は、プロセッサとコプロセッサとの同期を取りつつ、プロセッサの消費電力を削減する同期信号生成回路およびその方法を提供することである。

【0008】

第3の目的は、システム全体のオーバーヘッドを低減したプロセッサシステムを提供することである。

【0009】

【課題を解決するための手段】

請求項1に記載の同期信号生成回路は、プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路であって、共有メモリにおけるプロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、プロセッサがアクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力する第1の論理回路とを含む。

【0010】

第1の論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、その

オーバヘッドを低減することが可能となる。

【0011】

請求項2に記載の同期信号生成回路は、請求項1記載の同期信号生成回路であって、第1の論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号、比較回路の比較結果および共有メモリをロック状態にしないことを示す信号に基づいて、プロセッサをウェイト状態にする信号を出力する。

【0012】

したがって、共有メモリがロック状態であっても、プロセッサは共有メモリにアクセスすることができ、プロセッサおよびコプロセッサのデバッグが容易に行なえるようになる。

【0013】

請求項3に記載の同期信号生成回路は、請求項1または2記載の同期信号生成回路であって、同期信号生成回路はさらに、バスウェイトサイクル数をカウントするバスウェイトカウンタと、コプロセッサがコプロセッサ命令を実行中であることを示す信号、比較回路の比較結果およびバスウェイトカウンタのカウント値に基づいて、プロセッサにバスエラー信号を出力する第2の論理回路とを含む。

【0014】

第2の論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号、比較回路の比較結果およびバスウェイトカウンタのカウント値に基づいて、プロセッサにバスエラー信号を出力するので、プロセッサおよびコプロセッサの同期を取るときに誤ってバスアクセスエラーが発生することを防止することが可能となる。

【0015】

請求項4に記載の同期信号生成回路は、請求項1または2記載の同期信号生成回路であって、同期信号生成回路はさらに、第1の論理回路が出力する信号によってプロセッサがウェイト状態となるときのウェイト数を設定する第1の初期値レジスタと、プロセッサの通常アクセス時におけるウェイト数を設定する第2の初期値レジスタと、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、第1の初期値レジスタの値および第

2の初期値レジスタの値を選択して出力するセレクタと、セレクタによって選択された値を入力し、カウントしたウェイトサイクル数が選択された値となったときにプロセッサにバスエラー信号を出力するバスウェイトカウンタとを含む。

【0016】

したがって、プロセッサおよびコプロセッサの同期を取るときに発生するバスアクセスエラーと、プロセッサの通常アクセス時におけるバスアクセスエラーとのウェイトサイクル数を別々に設定することができ、誤ってバスアクセスエラーが発生することを防止することができる。

【0017】

請求項5に記載の同期信号生成回路は、請求項1～4のいずれかに記載の同期信号生成回路であって、同期信号生成回路はさらに、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサを低消費電力モードにする信号を出力する第3の論理回路とを含む。

【0018】

したがって、プロセッサの消費電力を削減することが可能となる。

請求項6に記載の同期信号生成回路は、プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路であって、共有メモリにおけるプロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、プロセッサがアクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持するコプロセッサ命令実行フラグと、コプロセッサ命令実行フラグに保持される情報および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力する論理回路とを含む。

【0019】

論理回路は、コプロセッサ命令実行フラグに保持される情報および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、そのオーバーヘッドを低減することが可能となる。

【0020】

請求項 7 に記載の同期信号生成回路は、請求項 6 記載の同期信号生成回路であって、アクセス禁止領域レジスタは、共有メモリにおける前記プロセッサの第 1 のアクセス禁止領域を指定する第 1 のアクセス禁止領域レジスタと、共有メモリにおけるプロセッサの第 2 のアクセス禁止領域を指定する第 2 のアクセス禁止領域レジスタとを含み、比較回路は、プロセッサが第 1 のアクセス禁止領域レジスタに指定された第 1 のアクセス禁止領域にアクセスしたことを検出する第 1 の比較回路と、プロセッサが前記第 2 のアクセス禁止領域レジスタに指定された第 2 のアクセス禁止領域にアクセスしたことを検出する第 2 の比較回路とを含み、コプロセッサ命令実行フラグは、コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持する第 1 のコプロセッサ命令実行フラグと、第 2 のコプロセッサ命令実行フラグとを含み、論理回路は、第 1 のコプロセッサ命令実行フラグに保持される情報および第 1 の比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力する第 1 の論理回路と、第 2 のコプロセッサ命令実行フラグに保持される情報および第 2 の比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力する第 2 の論理回路とを含む。

【 0 0 2 1 】

したがって、異なるアクセス禁止領域を設定することができ、コプロセッサがコプロセッサ命令を実行して多量の実行結果を順次共有メモリに格納する最中に、プロセッサのウェイト状態を解除してアクセス可能となった共有メモリの領域にプロセッサがアクセスすることが可能となる。

【 0 0 2 2 】

請求項 8 に記載のプロセッサシステムは、プロセッサと、コプロセッサと、プロセッサおよびコプロセッサに接続される共有メモリと、プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路とを含むプロセッサシステムであって、同期信号生成回路は、共有メモリにおけるプロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、プロセッサがアクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする

信号を出力する論理回路とを含む。

【 0 0 2 3 】

論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバヘッドを低減することが可能となる。

【 0 0 2 4 】

請求項 9 に記載のプロセッサシステムは、プロセッサと、コプロセッサと、プロセッサおよびコプロセッサに接続される共有メモリと、プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成回路とを含むプロセッサシステムであって、同期信号生成回路は、共有メモリにおけるプロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタと、プロセッサがアクセス禁止領域レジスタに指定されたアクセス禁止領域にアクセスしたことを検出する比較回路と、コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持するコプロセッサ命令実行フラグと、コプロセッサ命令実行フラグに保持される情報および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力する論理回路とを含む。

【 0 0 2 5 】

論理回路は、コプロセッサ命令実行フラグに保持される情報および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバヘッドを低減することが可能となる。

【 0 0 2 6 】

請求項 1 0 に記載の同期信号生成方法は、プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成方法であって、共有メモリにおけるプロセッサのアクセス禁止領域を指定するステップと、プロセッサが指定されたアクセス禁止領域にアクセスしたことを検出するステップと、コプロセッサがコプロセッサ命令を実行中であることを示す信号および検出結果に基づいて、プロセッサをウェイト状態にする信号を生成するステップとを含む。

【0027】

コプロセッサがコプロセッサ命令を実行中であることを示す信号および検出結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバヘッドを低減することが可能となる。

【0028】

請求項11に記載のプロセッサシステムは、プロセッサおよびコプロセッサによる共有メモリへのアクセスの同期を取る同期信号生成方法であって、共有メモリにおけるプロセッサのアクセス禁止領域を指定するステップと、プロセッサが指定されたアクセス禁止領域にアクセスしたことを検出するステップと、コプロセッサがコプロセッサ命令を実行中であることを示す情報を保持するステップと、保持される情報および検出結果に基づいて、プロセッサをウェイト状態にする信号を生成するステップとを含む。

【0029】

コプロセッサ命令実行フラグに保持される情報および検出結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバヘッドを低減することが可能となる。

【0030】

【発明の実施の形態】

(実施の形態1)

図1は、本発明の実施の形態1におけるプロセッサシステムの概略構成を示すブロック図である。このプロセッサシステム1は、プロセッサ11と、共有メモリ12aと、コプロセッサ13aとを含む。プロセッサ11、共有メモリ12aおよびコプロセッサ13aに同一のクロックが供給され、コプロセッサ13aがコプロセッサ命令を実行中には、プロセッサ11とコプロセッサ13aとが並行して処理を実行することになる。

【0031】

プロセッサ11が共有メモリ12aにアクセスする際に、アドレス(P_AD

DR) 信号、バススタート (P_BS) 信号およびリード/ライト (P_R/W) 信号を出力する。プロセッサ 11 が共有メモリ 12 a からデータを読み出す場合には、データバス (P_DATA) の値を入力する。また、プロセッサ 11 が共有メモリ 12 a にデータを書込む場合には、データバス (P_DATA) にデータを読み出す。なお、データレディ (P_DC) 信号は、プロセッサ 11 による共有メモリ 12 a へのアクセス時にデータが確定したことを示している。

【0032】

コプロセッサ 13 a が共有メモリ 12 a にアクセスする際に、アドレス (C_ADDR) 信号、バススタート (C_BS) 信号およびリード/ライト (C_R/W) 信号を出力する。コプロセッサ 13 a が共有メモリ 12 a からデータを読み出す場合には、データバス (C_DATA) の値を入力する。また、コプロセッサ 13 a が共有メモリ 12 a にデータを書込む場合には、データバス (C_DATA) にデータを読み出す。なお、データレディ (C_DC) 信号は、コプロセッサ 13 a による共有メモリ 12 a へのアクセス時にデータが確定したことを示している。また、コプロセッサ命令実行信号は、Hレベルのときにコプロセッサ 13 a がコプロセッサ命令を実行中であることを示している。

【0033】

共有メモリ 12 a は、デュアルポートメモリアレイ 14 と、プロセッサ 11 とコプロセッサ 13 a との間の同期を取るための P_DC 信号および C_DC 信号を生成する DC 生成回路 (同期信号生成回路) 15 a とを含む。

【0034】

図 2 は、プロセッサ 11 および共有メモリ 12 a の動作を説明するためのタイミングチャートである。このタイミングチャートは、プロセッサ 11 が共有メモリ 12 a からデータを読み出す場合 (通常のリードアクセス) を示している。バスサイクル “1” において、プロセッサ 11 は P_BS 信号を Hレベルにし、P_R/W 信号を Hレベルにするとともに、アドレス信号 P_ADDR にアクセスする共有メモリ 12 a のアドレス (A0) を出力する。共有メモリ 12 a は、アドレス信号 P_ADDR で指定された領域のデータ (D0) をデータバス P_DATA に出力するが、このサイクル内でデータが確定するため、共有メモリ 12 a

はこのサイクルでP__DC信号にHレベルを出力する。なお、バスサイクル“2”は、アイドルサイクルである。

【0035】

バスサイクル“3”において、プロセッサ11はP__BS信号をHレベルにし、P__R/W信号をHレベルにするとともに、アドレス信号P__ADDRにアクセスする共有メモリ12のアドレス(A1)を出力する。共有メモリ12aは、アドレス信号P__ADDRで指定された領域のデータ(D1)をデータバスP__DATAに出力するが、このサイクル内でデータが確定しないため、共有メモリ12aはこのサイクルでP__DC信号のLレベルを保持する。そして、バスサイクル“4”においてデータが確定するため、共有メモリ12aはこのサイクルでP__DC信号にHレベルを出力する。なお、バスサイクル“5”は、アイドルサイクルである。

【0036】

コプロセッサ13aが共有メモリ12aからデータを読出す場合も、プロセッサ11が共有メモリ12aからデータを読出す場合と同様のタイミングで動作が行なわれる。

【0037】

図3は、コプロセッサ13aの概略構成を示すブロック図である。コプロセッサ13はパイプライン処理によってコプロセッサ命令を実行するものであり、命令フェッチステージ21と、デコードステージ22と、実行ステージ23とを含む。なお、図3は、コプロセッサ命令実行信号の生成を説明するための図であり、それに関連する構成のみを記載したものである。たとえば、実行ステージ23のALU (Arithmetic and Logic Unit) 30は、アドレス演算のみを行なう構成となっており、データ演算に関連する構成は省略している。

【0038】

命令フェッチステージ21において、フェッチされた命令は命令レジスタ24に設定される。この命令レジスタ24の最上位ビットには、コプロセッサ命令ビットが割当てられており、このビットによってコプロセッサ命令であるか否かが検出される。命令レジスタ24の命令の種類を示すビット、ディスティネーショ

ン、ソースAおよびソースBは、デコードステージ22のデコーダ28に入力される。

【0039】

デコードステージ22において、コプロセッサ命令検出部25は、命令デコードのタイミングと同じタイミングでコプロセッサ命令ビットの判別を行ない、その命令がコプロセッサ命令であれば実行ステージ23のコプロセッサ命令実行信号生成部29へその旨を通知する。デコーダ28は、命令レジスタ24の命令の種類を示すビット、デスティネーション、ソースAおよびソースBをデコードし、そのデコード結果によって命令の種類を判別し、レジスタ26の選択や即値27の設定を行なう。

【0040】

実行ステージ23において、ステートマシン31は、デコーダ28のデコード結果によって命令の実行サイクル数を決定する。ステートマシン31は、命令の実行サイクルが終了するときに、その旨をコプロセッサ命令実行信号生成部29へ出力する。コプロセッサ命令実行信号生成部29は、コプロセッサ命令検出部25から通知を受けたときにコプロセッサ命令実行信号をHレベルにし、ステートマシン31から命令の実行サイクルの終了通知を受けたときにコプロセッサ命令実行信号をLレベルにする。なお、ALU30は、デコードステージ22のソースバスAおよびソースバスBの値を入力し、アドレス演算を行なうことによって、アクセスする共有メモリ12aのアドレスC_ADDRを出力したり、内部レジスタ32を選択したりする。

【0041】

図4は、コプロセッサ13aのコプロセッサ命令実行信号のタイミングを説明するための図である。サイクル“1”において命令がフェッチされ、サイクル“2”においてフェッチされた命令がデコードされる。このデコードの結果、コプロセッサ命令であると判定されると、コプロセッサ命令実行信号がHレベルとなる。そして、サイクル“3”以降でコプロセッサ命令が実行され、実行サイクルの終了時にコプロセッサ命令実行信号がLレベルとなる。

【0042】

図5は、プロセッサ11がコプロセッサ命令を発行してから、コプロセッサ命令の実行結果を取得するまでの処理を説明するための図である。まず、プロセッサ11がコプロセッサ命令をコプロセッサ13に発行すると（S1）、コプロセッサ13aはコプロセッサ命令の実行を開始し（S2）、プロセッサ11から共有メモリ12aへのメモリアクセスがロック状態となる（S3）。なお、プロセッサ11がコプロセッサ13aへコプロセッサ命令を発行する際の形態として、コプロセッサ13aがプロセッサ11と同じタイミングで命令をフェッチし、フェッチした命令がコプロセッサ命令であれば実行を開始する方法や、プロセッサ11がコプロセッサの命令メモリ領域にコプロセッサ命令を書込む方法等が挙げられるが、これらに限られるものではない。

【0043】

プロセッサ11がコプロセッサ命令の実行結果を取得するために、共有メモリ12aにアクセスすると（S4）、共有メモリ12aはプロセッサ11に対してLレベルのP__DC信号を出力し、ウェイト状態にする（S5）。

【0044】

コプロセッサ13aがコプロセッサ命令の実行を終了すると（S6）、プロセッサ11からのメモリアクセスのロック状態が解除され（S7）、共有メモリ12aはプロセッサ11に対してHレベルのP__DC信号を出力し、ウェイト状態を解除する（S8）。その結果、プロセッサ11は共有メモリ12aからコプロセッサ命令の実行結果を取得する。

【0045】

図6は、プロセッサ11から共有メモリ12aへのメモリアクセスがロック状態となっている場合のタイミングチャートである。サイクル“1”においてロック状態になると、プロセッサ11がコプロセッサ命令の実行結果が格納される共有メモリ12aの領域にアクセスしても、共有メモリ12aがP__DC信号にLレベルを出力したままであり、プロセッサ11はウェイト状態のままとなる。したがって、プロセッサ11は誤ったデータを読出すという誤動作が防止される。

【0046】

コプロセッサ13aがコプロセッサ命令の実行を終了すると、共有メモリ12

a のロック状態が解除され、共有メモリ 1 3 a は P__DC 信号を H レベルにする。その結果、共有メモリ 1 3 a の P__ADDR で指定された領域のデータが P__DATA に出力され、プロセッサ 1 1 がコプロセッサ命令の実行結果を讀出すことができる。

【0047】

図 7 は、プロセッサ 1 1 へ出力する P__DC 信号を生成する回路の一例を示す図である。この P__DC 生成回路は、プロセッサ 1 1 が共有メモリ 1 2 a に通常アクセスを行なったときに、0 ウェイトで動作する回路を示している。P__DC 生成回路は、アクセス禁止領域レジスタ 5 0 と、比較器 5 1 と、NAND 回路 5 3 と、AND 回路 5 4 とを含む。アクセス禁止領域レジスタ 5 0 には、コプロセッサ 1 3 がコプロセッサ命令を実行中に、プロセッサ 1 1 からのアクセスを禁止すべき共有メモリ 1 2 の領域のアドレスを示す値が設定される。比較回路 5 1 は、アクセス禁止領域レジスタ 5 0 に設定された値と、プロセッサ 1 1 が出力した P__ADDR の値とを比較し、一致していれば一致信号に H レベルを出力する。

【0048】

プロセッサ 1 1 がアクセス禁止領域レジスタ 5 0 に格納された領域にアクセスすると、比較器 5 1 からの一致信号 5 2 が H レベルとなる。そのとき、コプロセッサ 1 3 a がコプロセッサ命令を実行中であれば、コプロセッサ命令実行信号が H レベルとなるので、NAND 回路 5 3 の出力が L レベルとなる。その結果、AND 回路 5 4 の出力である P__DC 信号が L レベルとなって、プロセッサ 1 1 がウェイト状態となる。

【0049】

また、コプロセッサ 1 3 a がコプロセッサ命令の実行を終了すると、コプロセッサ命令実行信号が L レベルとなり、NAND 回路 5 3 の出力が H レベルとなる。P__BS 信号が H レベルであるので、AND 回路 5 4 の出力である P__DC 信号が H レベルとなって、プロセッサ 1 1 のウェイト状態が解除される。

【0050】

以上説明したように、本実施の形態におけるプロセッサシステムによれば、コプロセッサ 1 3 a がコプロセッサ命令を実行しているときは、そのコプロセッサ

命令の実行結果が格納される共有メモリ 1 2 a の領域に対するプロセッサ 1 1 のアクセスがロックされるので、プロセッサ 1 1 が誤った実行結果を読出すことを防止できる。また、コプロセッサ命令の実行が完了するまでプロセッサ 1 1 がウェイト状態となるだけであるので、ポーリングや割込み処理などのプロセッサ 1 1 による無駄な処理サイクルがなくなり、プロセッサ 1 1 の処理速度を向上させることができるとともに、消費電力を削減するが可能となった。

【 0 0 5 1 】

（実施の形態 2）

図 8 は、本発明の実施の形態 2 におけるプロセッサシステムの概略構成を示すブロック図である。このプロセッサシステム 2 の構成は、図 1 に示す実施の形態 1 におけるプロセッサシステムの構成と比較して、共有メモリおよびコプロセッサの内部構成が異なる点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。なお、本実施の形態における共有メモリおよびコプロセッサの参照符号をそれぞれ 1 2 b および 1 3 b とする。

【 0 0 5 2 】

本実施の形態 1 におけるコプロセッサ 1 3 b は、図 3 に示す実施の形態 1 におけるコプロセッサ 1 3 a と比較して、コプロセッサ命令実行信号を生成するための回路が削除された点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。

【 0 0 5 3 】

図 9 は、プロセッサ 1 1 へ出力する P _ D C 信号を生成する回路の一例を示す図である。この P _ D C 生成回路は、プロセッサ 1 1 が共有メモリ 1 2 b に通常アクセスを行なったときに、0 ウェイトで動作する回路を示している。P _ D C 生成回路は、アクセス禁止領域レジスタ 5 0 と、比較器 5 1 と、N A N D 回路 5 3 と、A N D 回路 5 4 と、コプロセッサ命令実行フラグ 5 5 とを含む。アクセス禁止領域レジスタ 5 0 には、コプロセッサ 1 3 がコプロセッサ命令を実行中に、プロセッサ 1 1 からのアクセスを禁止すべき共有メモリ 1 2 の領域のアドレスが設定される。比較回路 5 1 は、アクセス禁止領域レジスタ 5 0 に設定された値と、プロセッサ 1 1 が出力した P _ A D D R の値とを比較し、一致していれば一致

信号52にHレベルを出力する。

【0054】

コプロセッサ命令実行フラグ55は、コプロセッサ13bの命令によって書込まれるフラグであり、フリップフロップ等によって構成される。コプロセッサ命令実行フラグ55に書込まれる値は、コプロセッサ13bが命令を発行したときのデータバスC_DATAの値である。コプロセッサ13bがコプロセッサ命令を実行する際に、コプロセッサ命令実行フラグ55に“1”を設定する。そして、コプロセッサ13bがコプロセッサ命令の実行を完了したときにコプロセッサ命令実行フラグ55に“0”を設定する。

【0055】

プロセッサ11がアクセス禁止領域レジスタ50に格納された領域にアクセスすると、比較器51からの一致信号がHレベルとなる。そのとき、コプロセッサ13aがコプロセッサ命令実行フラグ55に“1”を設定していれば、NAND回路53の出力がLレベルとなる。その結果、AND回路54の出力であるP_DC信号がLレベルとなって、プロセッサ11がウェイト状態となる。

【0056】

また、コプロセッサ13aがコプロセッサ命令の実行を完了すると、コプロセッサ命令実行フラグ55に“0”を設定し、NAND回路53の出力がHレベルとなる。P_BS信号がHレベルであるので、AND回路54の出力であるP_DC信号がHレベルとなって、プロセッサ11のウェイト状態が解除される。

【0057】

コプロセッサ13bがコプロセッサ命令を実行する際に、実行結果が多量に存在して共有メモリ12bに多量のデータを書込む必要がある場合には、コプロセッサ命令の実行の進捗に伴ってロックする共有メモリ12bの領域を変化させることができる。この処理によって、既に実行結果が共有メモリ12bに書込まれている範囲内において、プロセッサ11の動作を先に進めることができる。この回路の一例を次に説明する。

【0058】

図10は、プロセッサ11からのアクセス禁止領域を2種類設定することがで

きるP__DC信号生成回路を示す図である。このP__DC生成回路は、プロセッサ11が共有メモリ12bに通常アクセスを行なったときに、0ウェイトで動作する回路を示している。P__DC生成回路は、アクセス禁止領域レジスタAおよびアクセス禁止領域レジスタB(60, 70)と、比較器61および71と、NAND回路63および73と、AND回路64および74と、コプロセッサ命令実行フラグAおよびコプロセッサ命令実行フラグB(65, 75)と、OR回路76とを含む。なお、図10に示す回路は、図9に示す回路を2つ用意し、それぞれのAND回路からの出力をOR回路76に入力するようにしたものである。

【0059】

アクセス禁止領域レジスタAおよびB(60, 70)にはそれぞれ、コプロセッサ13bがコプロセッサ命令を実行中に、プロセッサ11からのアクセスを禁止すべき共有メモリ12の異なる領域のアドレスが設定される。コプロセッサ13bは、コプロセッサ命令を実行する際に、アクセス禁止領域レジスタAおよびB(60, 70)のそれぞれに、アクセス禁止領域Aおよびアクセス禁止領域Bを設定する。

【0060】

コプロセッサ13bによるコプロセッサ命令の実行が進み、アクセス禁止領域Aまたはアクセス禁止領域Bのロックを解除できる状態になると、コプロセッサ13bはそれに対応するコプロセッサ命令実行フラグに“0”を設定する。その結果、一方の共有メモリ12bのロック状態が解除され、プロセッサ11がその領域にアクセス可能となる。コプロセッサ13bがコプロセッサ命令の実行を完了したときに、他方のコプロセッサ命令実行フラグに“0”を設定して、共有メモリ12bのロック状態を完全に解除する。

【0061】

以上説明したように、本実施の形態におけるプロセッサシステムによれば、コプロセッサ13bが命令を発行し、コプロセッサ命令実行フラグに値を設定して、プロセッサ11のアクセスをロックするようにしたので、プロセッサ11が誤った実行結果を讀出すことを防止できる。また、コプロセッサ命令の実行が完了するまでプロセッサ11がウェイト状態となるだけであるので、ポーリングや割

込み処理などのプロセッサ 1 1 による無駄な処理サイクルがなくなり、プロセッサ 1 1 の処理速度を向上させることができるとともに、消費電力を削減することが可能となった。

【 0 0 6 2 】

また、アクセス禁止領域を複数設定できるようにしたので、実行結果が多量に存在して共有メモリ 1 2 b に多量のデータを書込む必要がある場合には、既に実行結果が共有メモリ 1 2 b に書込まれている範囲内において、プロセッサ 1 1 の動作を先に進めることが可能となった。

【 0 0 6 3 】

(実施の形態 3)

一般に、プロセッサシステムにおいては、バスアクセスの異常時にバスアクセスエラー処理が行なわれる。バスアクセス異常の代表的な例として、プロセッサがメモリが存在しない領域にアクセスして DC 信号の応答がなく、プロセッサがハングアップしてしまうことが挙げられる。このバスアクセスエラー処理として、プロセッサがバスアクセス時に想定している以上のウェイトが挿入されたことを検出して以降の処理を行なうのが一般的である。

【 0 0 6 4 】

本発明の実施の形態 1 および 2 のプロセッサシステムにおいては、コプロセッサがコプロセッサ命令の実行を完了するまで、プロセッサからの共有メモリへのアクセスを制限し、その間 DC 信号が L レベルに保つことによりプロセッサをウェイト状態としている。したがって、上述した本来のバスアクセスエラーと、プロセッサとコプロセッサとの同期を取るために挿入したウェイト状態とが混乱し、プロセッサとコプロセッサとの同期を取っている最中にバスアクセスエラーと誤って認識される可能性がある。本実施の形態におけるプロセッサシステムは、このバスアクセスエラーの誤認識を防止したものである。

【 0 0 6 5 】

図 1 1 は、本発明の実施の形態 3 におけるプロセッサシステムの概略構成を示すブロック図である。このプロセッサシステム 3 の構成は、図 1 に示す実施の形態 1 におけるプロセッサシステムの構成と比較して、共有メモリ 1 2 c 内に B E

R R生成回路 4 2 が追加された点および P__B E R R 信号がプロセッサ 1 1 に入力される点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。

【 0 0 6 6 】

図 1 2 は、B E R R 生成回路 4 2 の構成を示す図である。この B E R R 生成回路 4 2 は、アクセス禁止領域レジスタ 5 0 と、比較器 5 1 と、N A N D 回路 5 3 と、バスウェイトカウンタ 8 0 と、A N D 回路 8 2 とを含む。バスウェイトカウンタ 8 0 は、ウェイトサイクル数をカウントし、オーバフローしたときにバスウェイトカウンタオーバフロー信号に H レベルを出力する。

【 0 0 6 7 】

プロセッサ 1 1 がアクセス禁止領域レジスタ 5 0 に格納された領域にアクセスすると、比較器 5 1 からの一致信号 5 2 が H レベルとなる。そのとき、コプロセッサ 1 3 a がコプロセッサ命令を実行中であれば、コプロセッサ命令実行信号が H レベルとなるので、N A N D 回路 5 3 の出力が L レベルとなる。その結果、バスウェイトカウンタ 8 0 の値に関係なく、A N D 回路 8 2 の出力である P__B E R R 信号が L レベルとなって、バスアクセスエラー処理は行なわれない。

【 0 0 6 8 】

また、コプロセッサ 1 3 a がコプロセッサ命令の実行を終了すると、コプロセッサ命令実行信号が L レベルとなり、N A N D 回路 5 3 の出力が H レベルとなるが、バスウェイトカウンタ 8 0 は L レベルとなっているので P__B E R R 信号は L レベルのままとなる。また、コプロセッサ 1 3 a がコプロセッサ命令を実行していない場合には N A N D 回路 5 3 の出力は H レベルとなり、バスウェイトカウンタ 8 0 がオーバフローしたときに P__B E R R 信号が H レベルとなる。したがって、通常のバスアクセスエラーをプロセッサ 1 1 に通知することができる。

【 0 0 6 9 】

図 1 3 は、B E R R 生成回路 4 2 の構成の他の一例を示す図である。この B E R R 生成回路 4 2 は、アクセス禁止領域レジスタ 5 0 と、比較器 5 1 と、バスウェイトカウンタ 8 0 と、A N D 回路 9 0 と、初期値レジスタ A および B (9 2 , 9 3) と、セクタ 9 4 とを含む。バスウェイトカウンタ 8 0 は、ウェイトサイ

クル数をカウントし、オーバーフローしたときにバスウェイトカウンタオーバーフロー信号にHレベルを出力する。

【0070】

初期値レジスタA（92）は、コプロセッサ13aがコプロセッサ命令を実行しているときにバスウェイトカウンタ80に設定される初期値を格納する。また、初期値レジスタB（93）は、プロセッサ11による通常のアクセス時にバスウェイトカウンタ80に設定される初期値を格納する。

【0071】

プロセッサ11がアクセス禁止領域レジスタ50に格納されたアドレスに対応する領域にアクセスすると、比較器51からの一致信号52がHレベルとなる。そのとき、コプロセッサ13aがコプロセッサ命令を実行中であれば、コプロセッサ命令実行信号がHレベルとなり、初期値選択信号がHレベルとなるので、セレクタ94は初期値レジスタA（92）の出力を選択してバスウェイトカウンタ80に設定する。バスカウンタ80は、ウェイトサイクル数をカウントし、初期値レジスタA（92）に設定された値を超えたときにP_BERR信号にHレベルを出力する。

【0072】

また、コプロセッサ13aがコプロセッサ命令の実行を終了すると、コプロセッサ命令実行信号がLレベルとなり、初期値選択信号がLレベルとなるので、セレクタ94は初期値レジスタB（93）の出力を選択してバスウェイトカウンタ80に設定する。バスカウンタ80は、ウェイトサイクル数をカウントし、初期値レジスタB（93）に設定された値を超えたときにP_BERR信号にHレベルを出力する。

【0073】

以上説明したように、本実施の形態におけるプロセッサシステムによれば、コプロセッサ13aがコプロセッサ命令を実行している間は、P_BERR信号にLレベルが出力されるようにしたので、プロセッサ11とコプロセッサ13aとの同期を取るときに発生するウェイトによって、誤ってバスアクセスエラーが発生するのを防止することが可能となった。

【 0 0 7 4 】

また、コプロセッサ 1 3 a がコプロセッサ命令を実行している時に発生するウェイトサイクル数の初期値と、プロセッサ 1 1 による通常バスアクセス時におけるウェイトサイクル数の初期値とを別々に設定できるようにしたので、個々に適切な値を設定することによって誤ってバスアクセスエラーが発生することを防止することが可能となった。

【 0 0 7 5 】

(実施の形態 4)

本発明の実施の形態 1 ～ 3 のプロセッサシステムにおいては、コプロセッサがコプロセッサ命令の実行を完了するまで、プロセッサからの共有メモリへのアクセスを制限し、その間 DC 信号を L レベルに保つことによりプロセッサをウェイト状態としている。しかし、その間プロセッサ 1 1 にクロック信号が供給されているため、無駄な電力が消費されていることになる。本実施の形態におけるプロセッサシステムは、プロセッサ 1 1 へ P _ S T O P 信号を出力することによって、プロセッサ 1 1 を低消費電力モードにするものである。なお、低消費電力モードとして、プロセッサ 1 1 へのクロック供給を停止するのが代表的であるが、これに限られるものではない。

【 0 0 7 6 】

図 1 4 は、本発明の実施の形態 4 におけるプロセッサシステムの概略構成を示すブロック図である。このプロセッサシステム 4 の構成は、図 1 に示す実施の形態 1 におけるプロセッサシステムの構成と比較して、共有メモリ 1 2 d 内に P _ S T O P 生成回路 4 4 が追加された点および P _ S T O P 信号がプロセッサ 1 1 に入力される点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。

【 0 0 7 7 】

図 1 5 は、P _ S T O P 生成回路 4 4 の構成を示す図である。この P _ S T O P 生成回路 4 4 は、アクセス禁止領域レジスタ 5 0 と、比較器 1 0 1 と、AND 回路 1 0 3 とを含む。プロセッサ 1 1 がアクセス禁止領域レジスタ 5 0 に格納されたアドレスに対応する領域にアクセスすると、比較器 1 0 1 からの一致信号 5

2がHレベルとなる。そのとき、コプロセッサ13aがコプロセッサ命令を実行中であれば、コプロセッサ命令実行信号がHレベルとなるので、AND回路103の出力がHレベルとなる。その結果、AND回路103の出力であるP_STOP信号がHレベルとなって、プロセッサ11が低消費電力モードとなる。

【0078】

以上説明したように、本実施の形態におけるプロセッサシステムによれば、コプロセッサ13aがコプロセッサ命令を実行しているときに、プロセッサ11がアクセス禁止領域にアクセスすると、プロセッサ11がウェイト状態になるとともに、低消費電力モードになるようにしたので、実施の形態1において説明した効果に加えて消費電力をさらに削減するが可能となった。

【0079】

(実施の形態5)

本発明の実施の形態1～4のプロセッサシステムにおいては、コプロセッサがコプロセッサ命令の実行を完了するまで、プロセッサからの共有メモリへのアクセスを制限し、その間DC信号をLレベルに保つことによりプロセッサをウェイト状態としている。しかし、プロセッサおよびコプロセッサのデバッグ時に、共有メモリのアクセス禁止領域のデータ値の変化状況が把握できれば、プログラムデバッグ時に有効である。本実施の形態におけるプロセッサシステムは、コプロセッサがLD_NOLOCK命令を実行して、LD_NOLOCK命令実行信号を出力するものである。

【0080】

なお、このLD_NOLOCK命令実行信号を生成する回路は、図3に示すコプロセッサ命令検出部25およびコプロセッサ命令実行信号生成部29と同様に構成することができる。すなわち、コプロセッサ命令検出部25の代わりに、デコーダ28によってLD_NOLOCK命令が検出されたときにその旨を通知する回路をデコードステージ22内に設ける。そして、実行ステージ23内のコプロセッサ命令実行信号生成部29と同様の回路によって、LD_NOLOCK命令が検出された通知を受けたときにLD_NOLOCK信号をLレベルにし、ステートマシン31から実行サイクルの終了通知を受けたときにLD_NOLOC

K信号をHレベルにする。

【0081】

図16は、本発明の実施の形態5におけるプロセッサシステムの概略構成を示すブロック図である。このプロセッサシステム5の構成は、図1に示す実施の形態1におけるプロセッサシステムの構成と比較して、DC生成回路およびコプロセッサ13の構成が異なる点のみが異なる。したがって、重複する構成および機能の詳細な説明は繰返さない。

【0082】

図17は、プロセッサ11へ出力するP_DC信号を生成する回路の一例を示す図である。このP_DC生成回路は、プロセッサ11が共有メモリ12eに通常アクセスを行なったときに、0ウェイトで動作する回路を示している。P_DC生成回路は、アクセス禁止領域レジスタ50と、比較器51と、NAND回路53と、インバータ110と、AND回路111とを含む。

【0083】

プロセッサ11がアクセス禁止領域レジスタ50に格納されたアドレスに対応する領域にアクセスすると、比較器51からの一致信号52がHレベルとなる。そのとき、コプロセッサ13aがコプロセッサ命令を実行中であれば、コプロセッサ命令実行信号がHレベルとなるので、NAND回路53の出力がLレベルとなる。しかし、そのコプロセッサ命令がLD_NOLOCK命令であれば、LD_NOLOCK命令実行信号がHレベルとなるためOR回路110の出力がHレベルとなり、P_BS信号がHレベルとなっている。その結果、AND回路111の出力であるP_DC信号がHレベルとなって、プロセッサ11がウェイト状態とならない。

【0084】

以上説明したように、本実施の形態におけるプロセッサシステムによれば、コプロセッサ13eに共有メモリ12eがロック状態であってもアクセスを許容する命令を持たせたので、プロセッサ11が共有メモリ12eにアクセスできるようになり、プロセッサおよびコプロセッサのプログラムデバッグを容易に行なえるようになった。

【 0 0 8 5 】

今回開示された実施の形態は、すべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【 0 0 8 6 】

【発明の効果】

請求項 1 に記載の同期信号生成回路によれば、第 1 の論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、そのオーバーヘッドを低減することが可能となった。

【 0 0 8 7 】

請求項 2 に記載の同期信号生成回路によれば、共有メモリがロック状態であっても、プロセッサは共有メモリにアクセスすることができ、プロセッサおよびコプロセッサのデバッグが容易に行なえるようになった。

【 0 0 8 8 】

請求項 3 に記載の同期信号生成回路によれば、第 2 の論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号、比較回路の比較結果およびバスウェイトカウンタのカウント値に基づいて、プロセッサにバスエラー信号を出力するので、プロセッサおよびコプロセッサの同期を取るときに誤ってバスアクセスエラーが発生することを防止することが可能となった。

【 0 0 8 9 】

請求項 4 に記載の同期信号生成回路によれば、プロセッサおよびコプロセッサの同期を取るときに発生するバスアクセスエラーと、プロセッサの通常アクセス時におけるバスアクセスエラーとのウェイトサイクル数を別々に設定することができ、誤ってバスアクセスエラーが発生することを防止することができた。

【 0 0 9 0 】

請求項 5 に記載の同期信号生成回路によれば、プロセッサの消費電力を削減す

ることが可能となった。

【0091】

請求項6に記載の同期信号生成回路によれば、論理回路は、コプロセッサ命令実行フラグに保持される情報および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、そのオーバーヘッドを低減することが可能となった。

【0092】

請求項7に記載の同期信号生成回路によれば、異なるアクセス禁止領域を設定することができ、コプロセッサがコプロセッサ命令を実行して多量の実行結果を順次共有メモリに格納する最中に、プロセッサのウェイト状態を解除してアクセス可能となった共有メモリの領域にプロセッサがアクセスすることが可能となった。

【0093】

請求項8に記載のプロセッサシステムによれば、論理回路は、コプロセッサがコプロセッサ命令を実行中であることを示す信号および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバーヘッドを低減することが可能となった。

【0094】

請求項9に記載のプロセッサシステムによれば、論理回路は、コプロセッサ命令実行フラグに保持される情報および比較回路の比較結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバーヘッドを低減することが可能となった。

【0095】

請求項10に記載の同期信号生成方法によれば、コプロセッサがコプロセッサ命令を実行中であることを示す信号および検出結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバーヘッドを低減することが可能とな

った。

【0096】

請求項11に記載の同期信号生成方法によれば、コプロセッサ命令実行フラグに保持される情報および検出結果に基づいて、プロセッサをウェイト状態にする信号を出力するので、プロセッサによる無駄な処理サイクルが不要となり、プロセッサシステム全体のオーバヘッドを低減することが可能となった。

【図面の簡単な説明】

【図1】 本発明の実施の形態1におけるプロセッサシステムの概略構成を示すブロック図である。

【図2】 プロセッサ11および共有メモリ12aの動作を説明するためのタイミングチャートである。

【図3】 コプロセッサ13aの概略構成を示すブロック図である。

【図4】 コプロセッサ13aのコプロセッサ命令実行信号のタイミングを説明するための図である。

【図5】 プロセッサ11がコプロセッサ命令を発行してから、コプロセッサ命令の実行結果を取得するまでの処理を説明するための図である。

【図6】 プロセッサ11から共有メモリ12aへのメモリアクセスがロック状態となっている場合のタイミングチャートである。

【図7】 プロセッサ11へ出力するP_DC信号を生成する回路の一例を示す図である。

【図8】 本発明の実施の形態2におけるプロセッサシステムの概略構成を示すブロック図である。

【図9】 プロセッサ11へ出力するP_DC信号を生成する回路の一例を示す図である。

【図10】 プロセッサ11からのアクセス禁止領域を2種類設定することができるP_DC信号生成回路を示す図である。

【図11】 本発明の実施の形態3におけるプロセッサシステムの概略構成を示すブロック図である。

【図12】 BERR生成回路42の構成の一例を示す図である。

【図 1 3】 B E R R 生成回路 4 2 の構成の他の一例を示す図である。

【図 1 4】 本発明の実施の形態 4 におけるプロセッサシステムの概略構成を示すブロック図である。

【図 1 5】 P _ S T O P 生成回路 4 4 の構成の一例を示す図である。

【図 1 6】 本発明の実施の形態 5 におけるプロセッサシステムの概略構成を示すブロック図である。

【図 1 7】 プロセッサ 1 1 へ出力する P _ D C 信号を生成する回路の一例を示す図である。

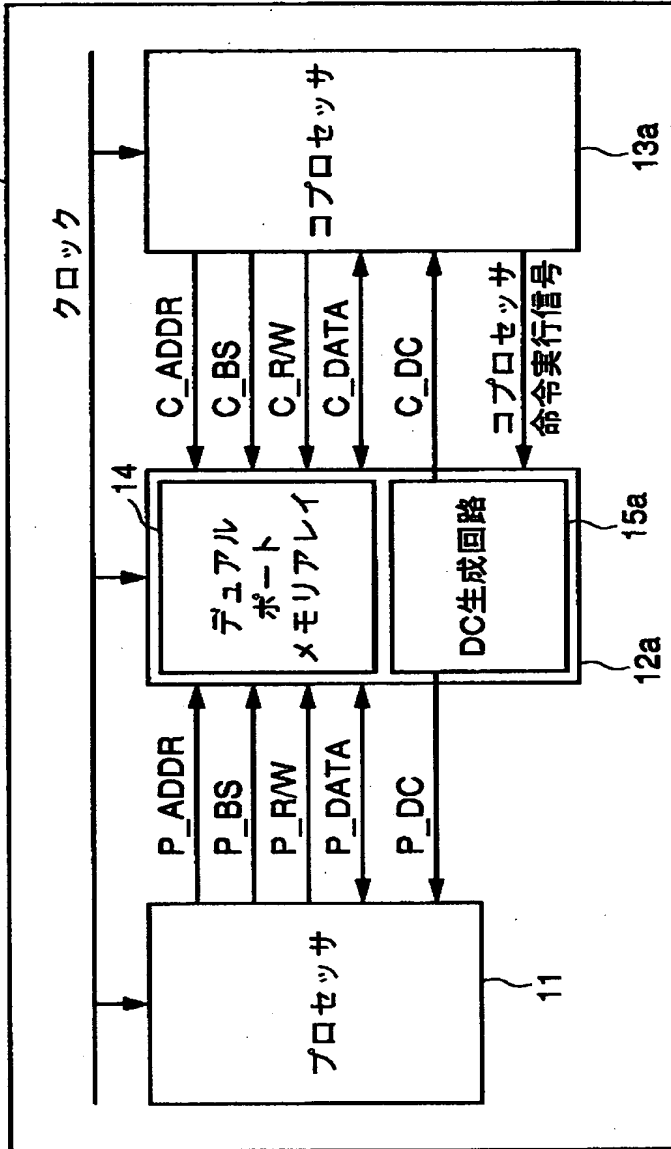
【符号の説明】

1 ~ 5 プロセッサシステム、1 1 プロセッサ、1 2 a, 1 2 b, 1 2 c, 1 2 d, 1 2 e 共有メモリ、1 3 a, 1 3 b コプロセッサ、1 4 デュアルポートメモリアレイ、1 5, 1 5 b, 1 5 e D C 生成回路、2 1 命令フェッチステージ、2 2 デコードステージ、2 3 実行ステージ、4 2 B E R R 生成回路、4 4 P _ S T O P 生成回路、5 0, 6 0, 7 0 アクセス禁止領域レジスタ、5 1, 6 1, 7 1 比較回路、5 5, 6 5, 7 5 コプロセッサ命令実行フラグ、8 0 バスウェイトカウンタ、9 2, 9 3 初期値レジスタ。

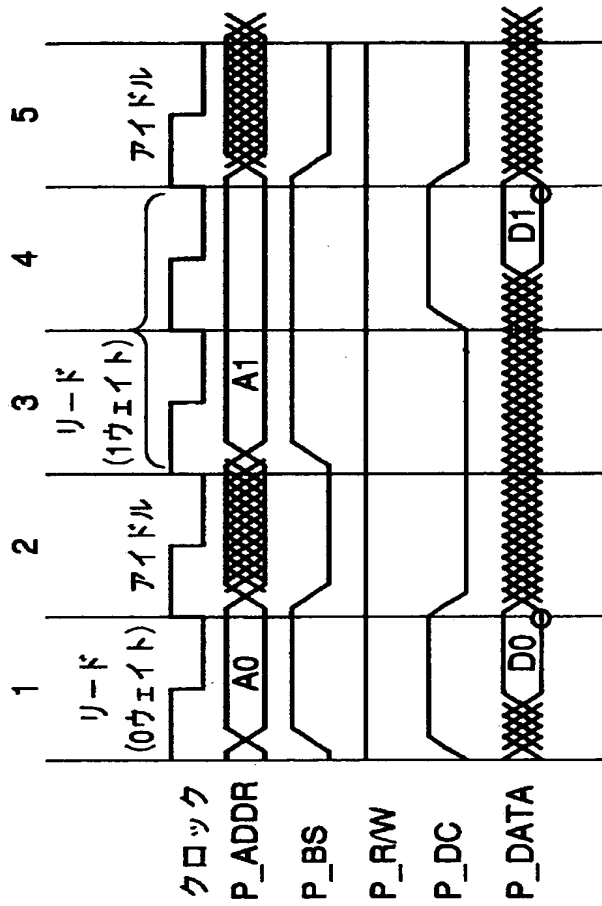
【書類名】

図面

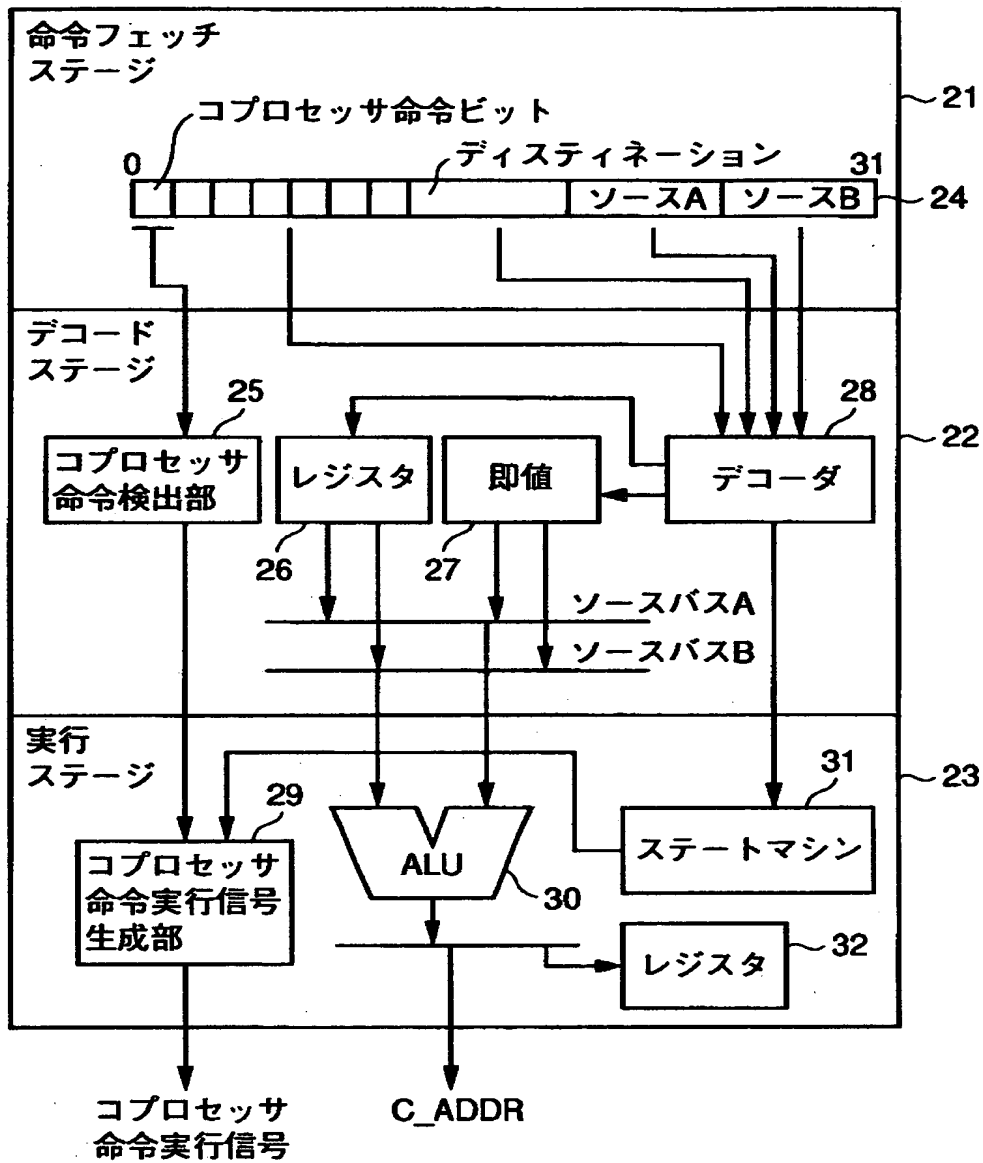
【図1】



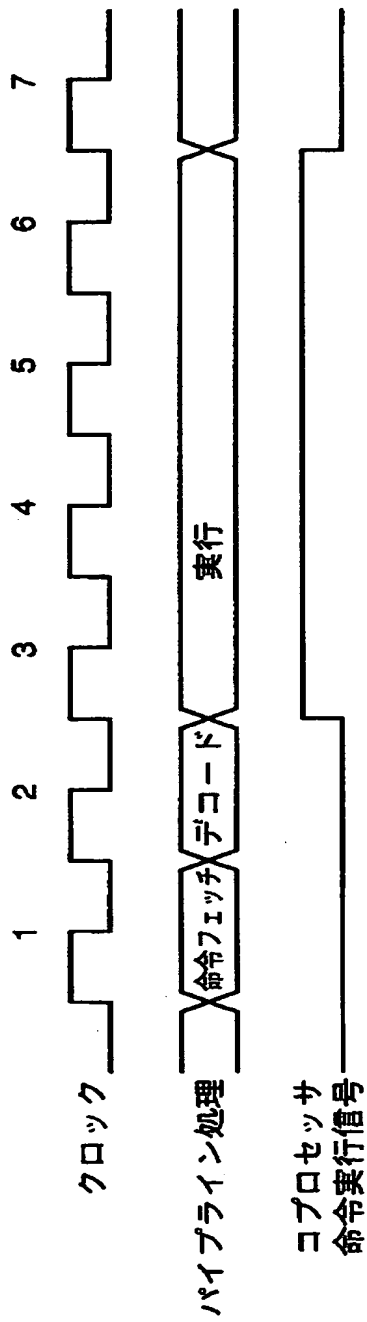
【図 2】



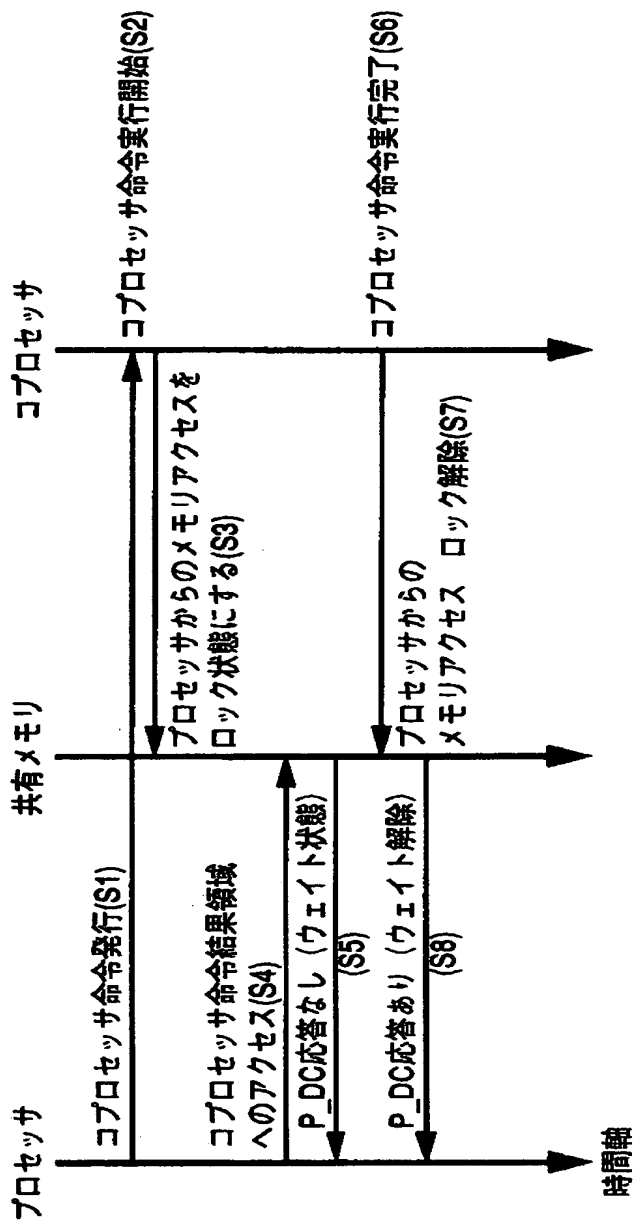
【図 3】



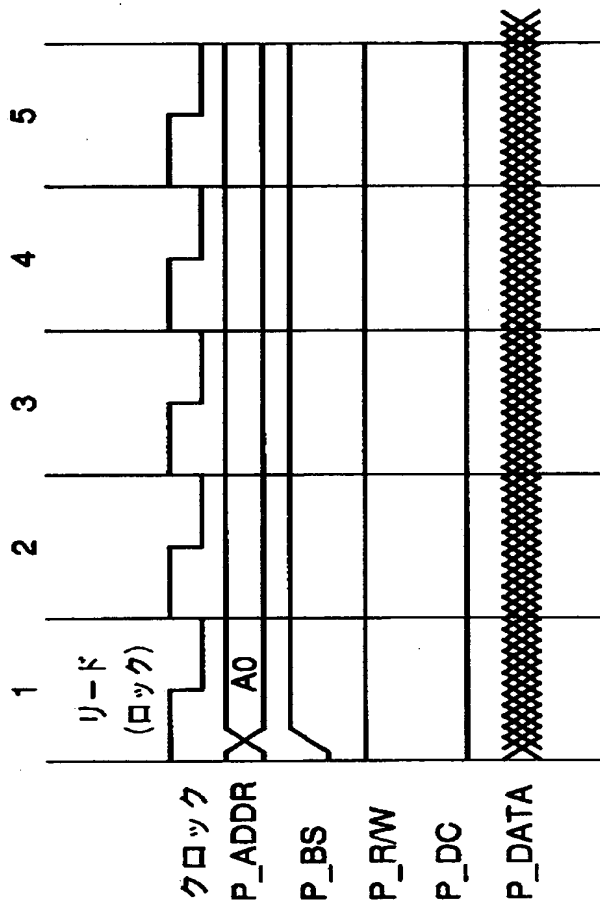
【図4】



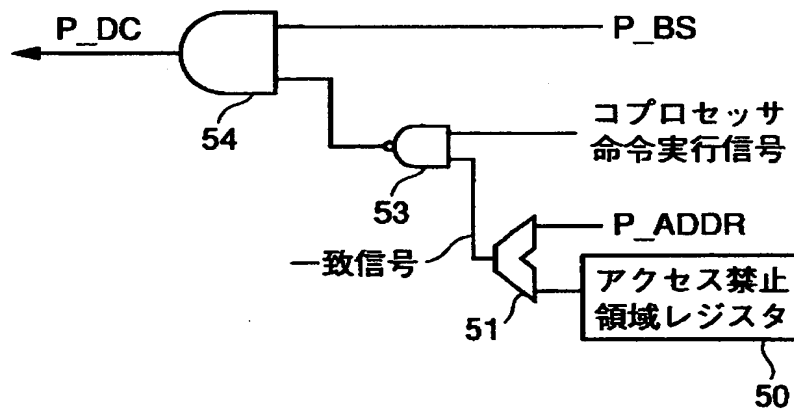
【図 5】



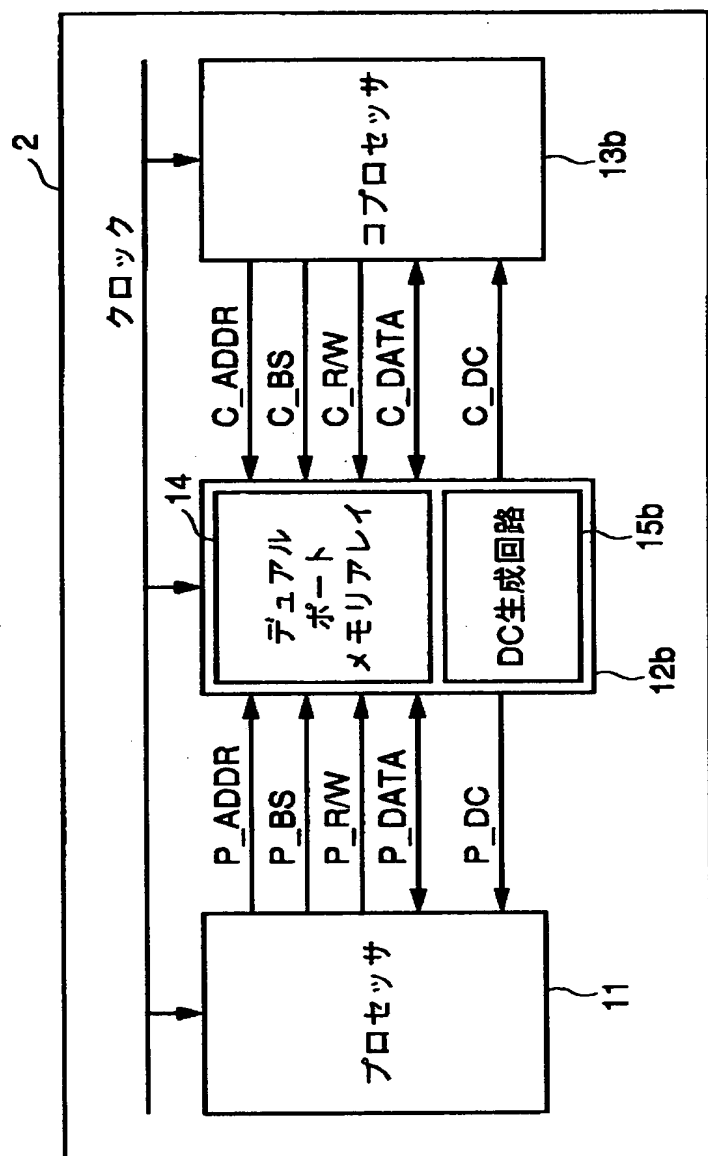
【図 6】



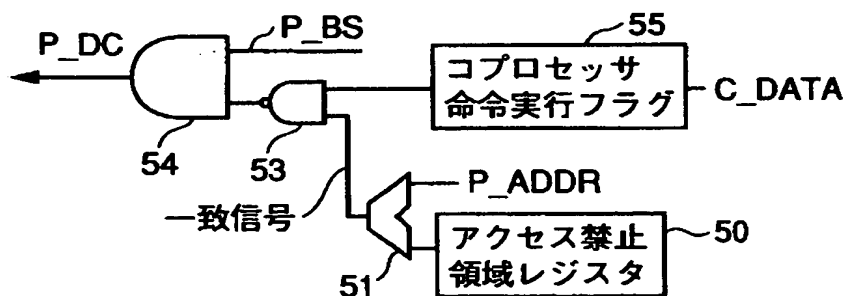
【図 7】



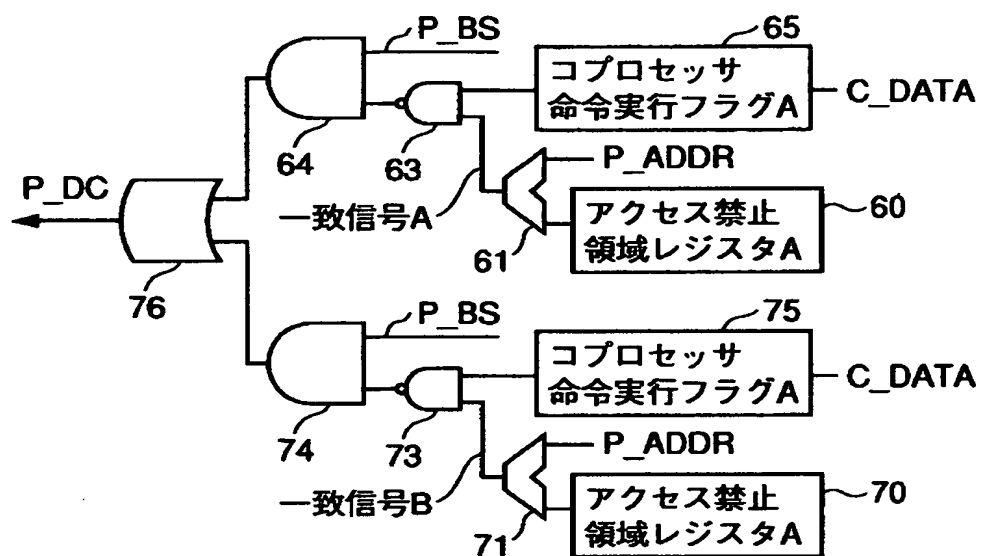
【図8】



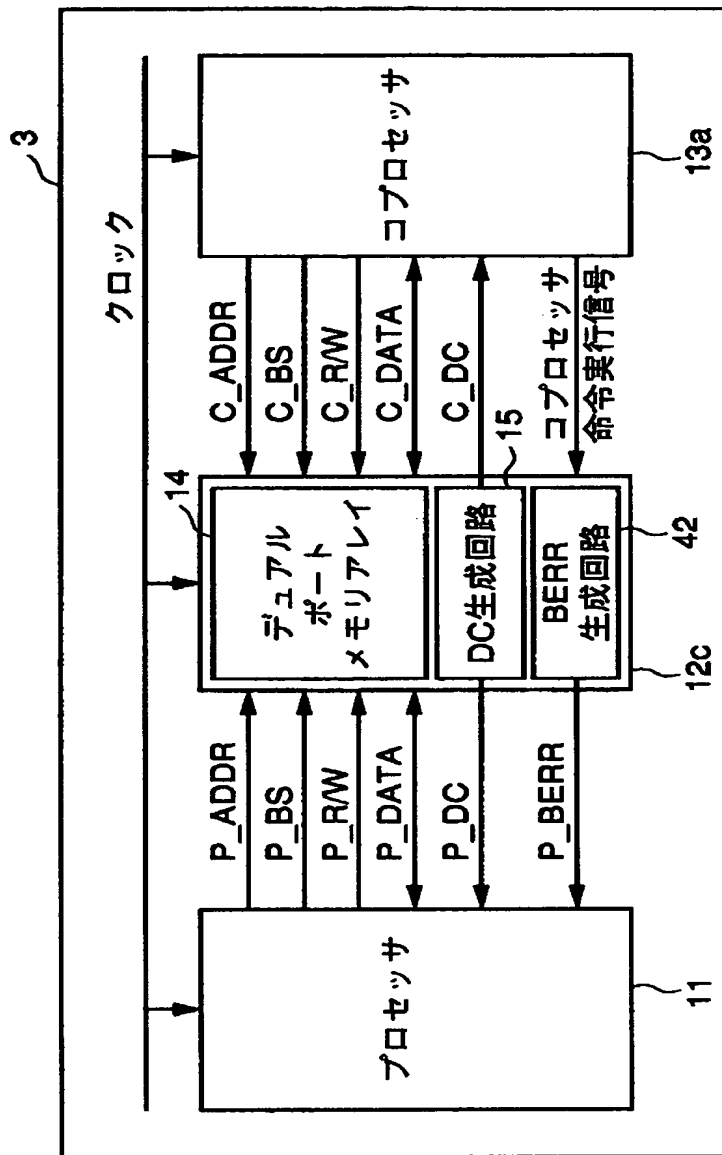
【図9】



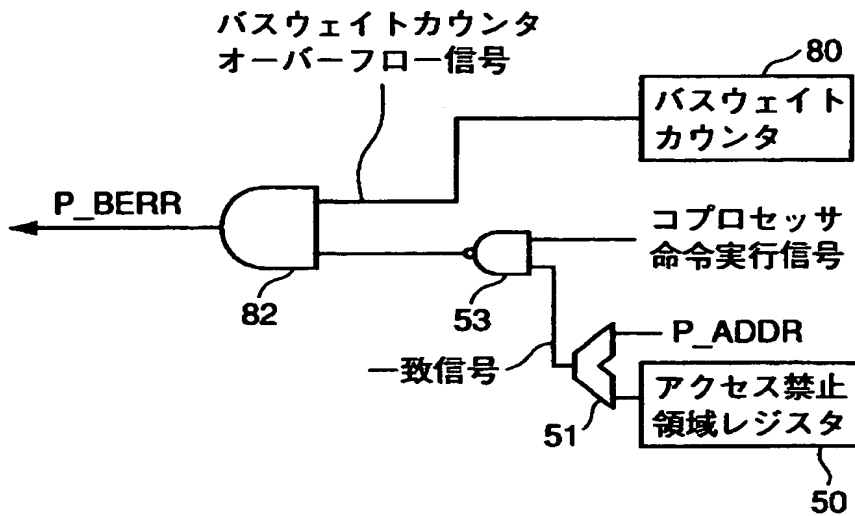
【図10】



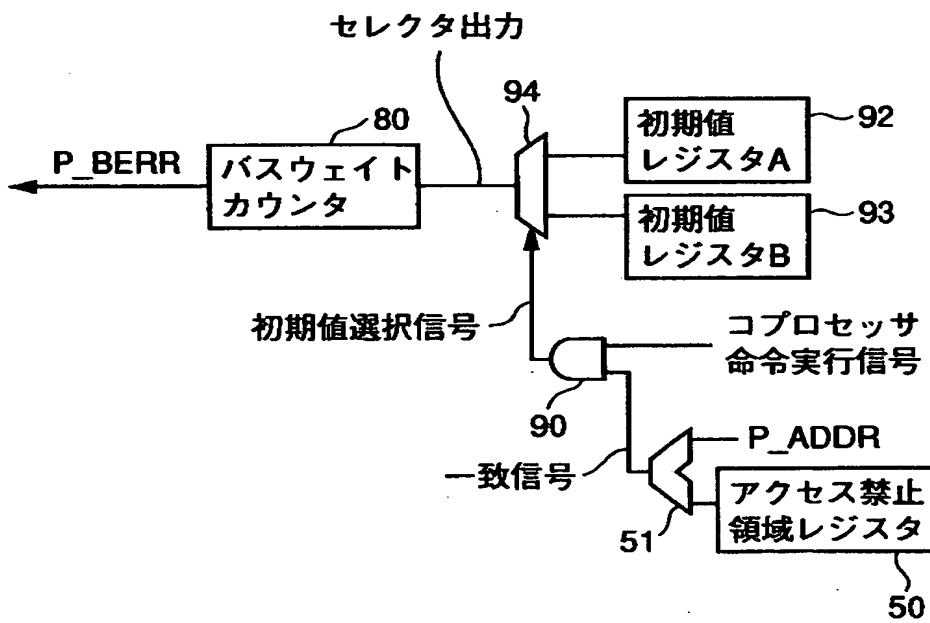
【図 11】



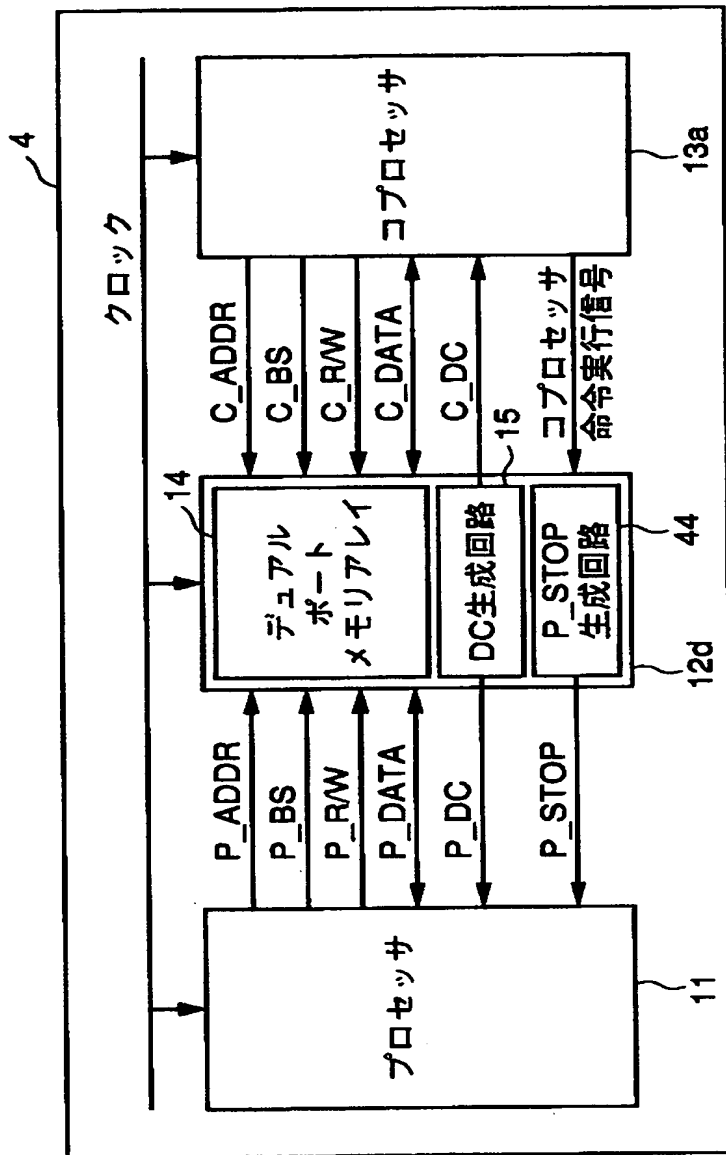
【図 1 2】



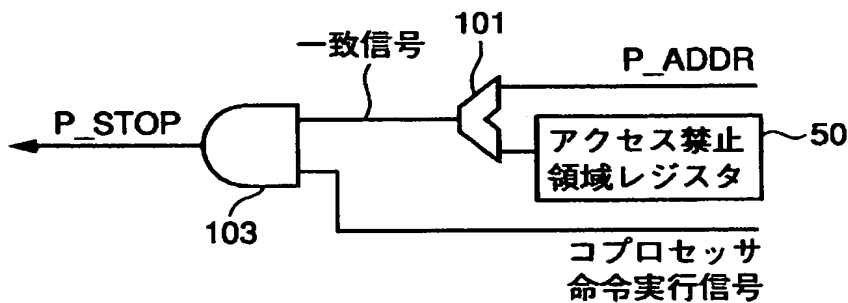
【図 1 3】



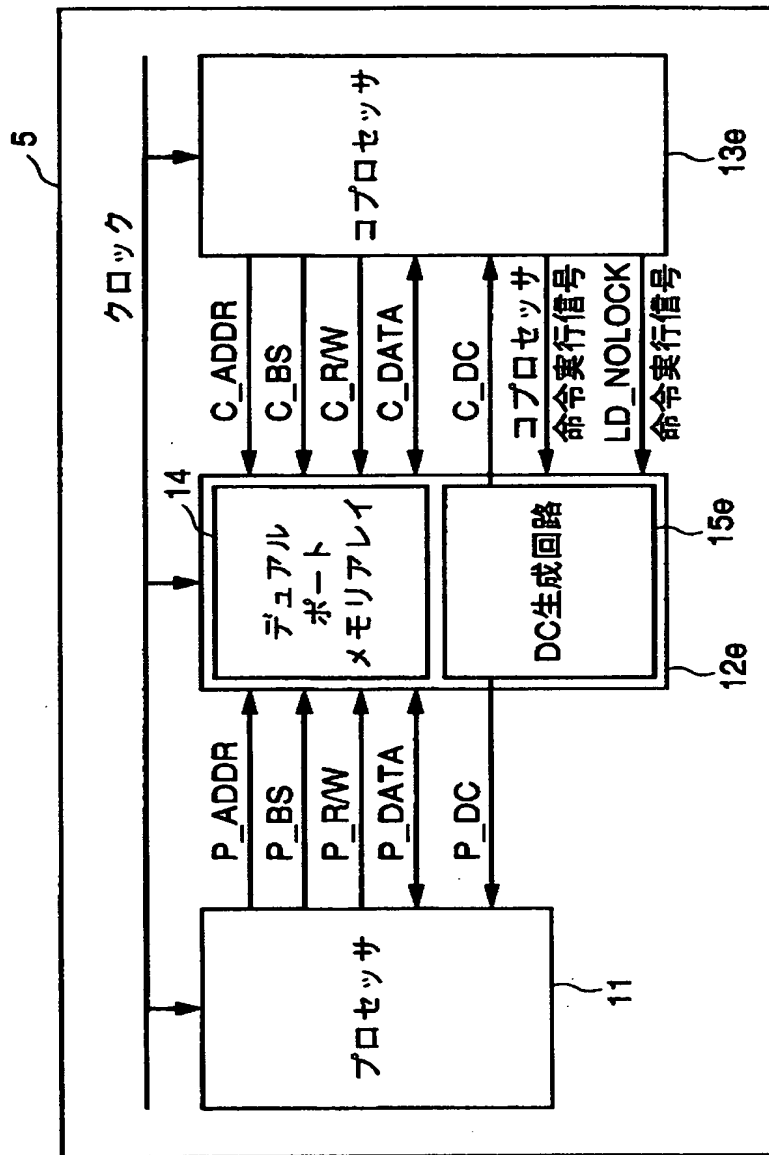
【図 14】



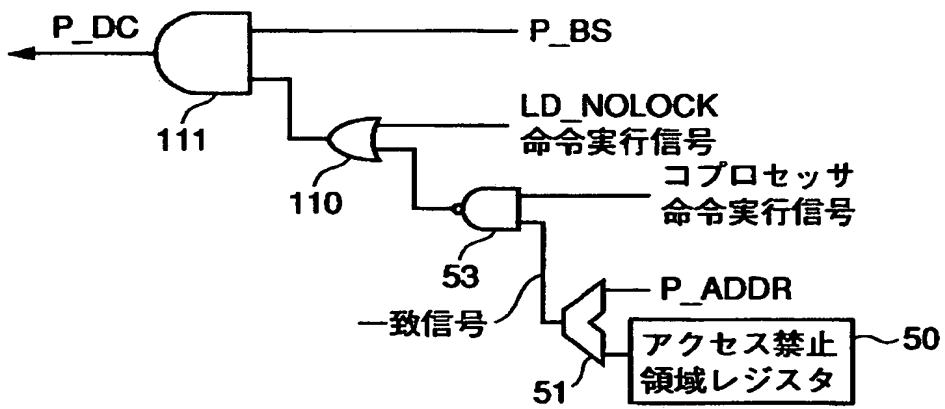
【図 15】



【図 16】



【図 17】



【書類名】 要約書

【要約】

【課題】 プロセッサとコプロセッサとの同期を取るときのオーバーヘッドを低減した同期信号生成回路を提供すること。

【解決手段】 同期信号生成回路は、共有メモリにおけるプロセッサのアクセス禁止領域を指定するアクセス禁止領域レジスタ50と、プロセッサがアクセス禁止領域レジスタ50に指定されたアクセス禁止領域にアクセスしたことを検出する比較回路51と、コプロセッサ命令実行信号および比較回路51の比較結果に基づいて、プロセッサをウェイト状態にするP_DC信号を生成するNAND回路53およびAND回路54とを含む。したがって、プロセッサによる無駄な処理サイクルが不要となり、そのオーバーヘッドを低減することが可能となる。

【選択図】 図7

出 願 人 履 歴 情 報

識別番号 {000006013}

1. 変更年月日	1990年 8月24日
[変更理由]	新規登録
住 所	東京都千代田区丸の内2丁目2番3号
氏 名	三菱電機株式会社